AWS - Install and Configure CLI

Lesson Downloads Cloud Resources

The AWS Command Line Interface (AWS CLI) is a command-line tool that allows you to interact with AWS services using commands in your terminal/command prompt.

AWS CLI enables you to run commands to provision, configure, list, delete resources in the AWS cloud. Before you run any of the **aws commands**, you need to follow three steps:

1. Install AWS CLI
2. Create an IAM user with Administrator permissions
3. Configure the AWS CLI

Step 1. Install AWS CLI v2

Refer to the official **AWS instructions to install/update AWS CLI** (version 2) based on your underlying OS. You can verify the installation using the following command in your terminal (macOS)/cmd (Windows).

```
# Display the folder that contains the symlink to the aws cli tool
which aws
# See the current version
aws --version
```

See the sample output below. Note that the exact version of AWS CLI and Python may vary in your system.

Mac/Linux/Windows: Verify the successful installation of AWS CLI 2

Show Transcript Summarize Video

Important

- **Note 1:** The default region is not always **us-east-2** as shown on the video. You need to set the default region to the region of your AWS account. To do this, you need to take note of the region next to your user account on the AWS console:
- **Note 2**: The **AdministratorAccess** permissions policy you attach with the IAM user will automatically be removed after the session ends in ~4 hours. In that case, you will encounter permission-related error while running the aws commands in your terminal. To fix the issue, refresh the classroom page, and atttach the IAM permissions policy again to the IAM user.

Region name of your AWS account

And then find the appropriate **Code** from the region mapping on this page: **AWS: Regions and Zones**.

Step 2. Create an IAM user

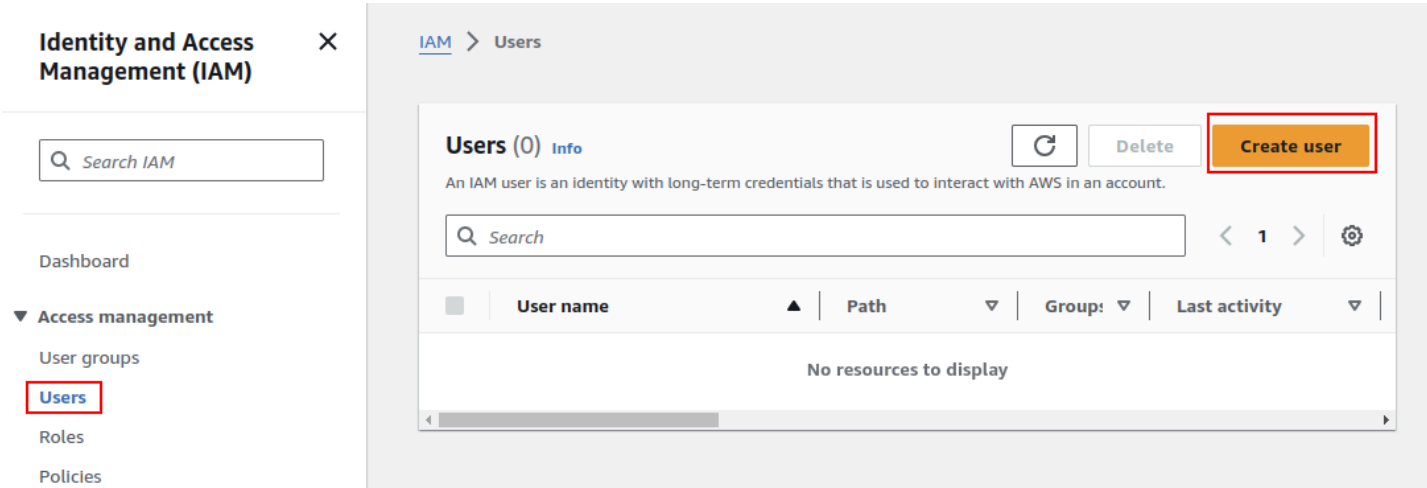
In this step, you will create an IAM user with Administrator permissions who is allowed to perform *any* action in your AWS account, only through CLI. After creating such an IAM user, we will use its **Access key** (long-term credentials)** **to configure the AWS CLI locally.

Let's create an **AWS IAM** user, and copy its Access key.

> AWS Identity and Access Management (IAM) service allows you to authorize users / applications (such as AWS CLI) to access AWS resources.

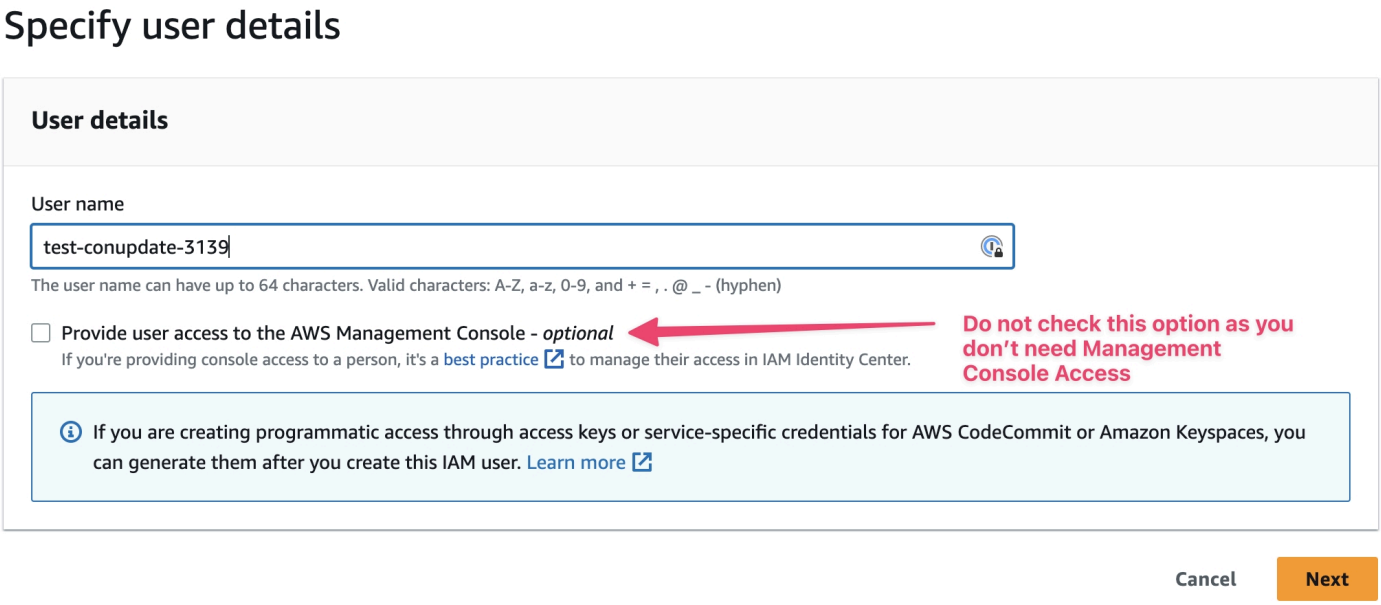
The Access key is a combination of an **Access Key ID** and a **Secret Access Key.** Let's see the steps to create an IAM user, and generate its Access key.

- Navigate to the **IAM Dashboard**, and create an IAM user.



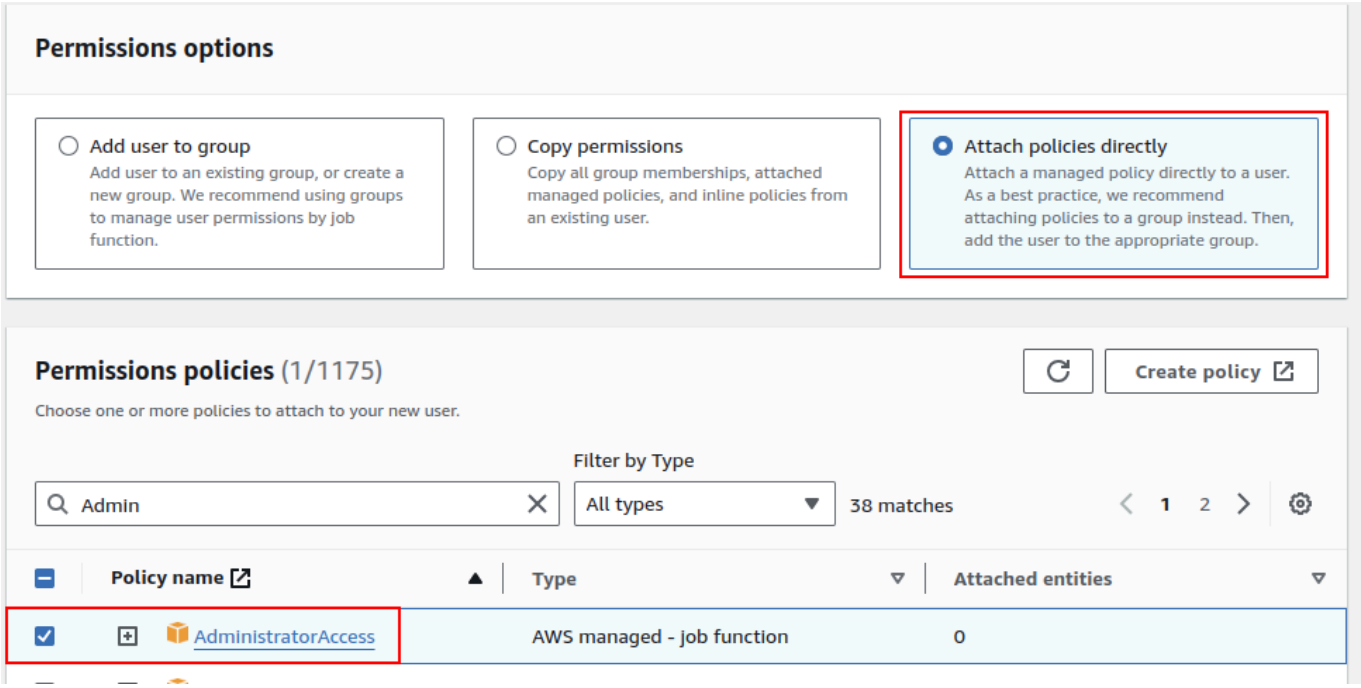
Add a new IAM user

- Set the user name, and click **Next**. **DO NOT** check **Provide user access to the AWS Management Console - *optional***.



Set User name.

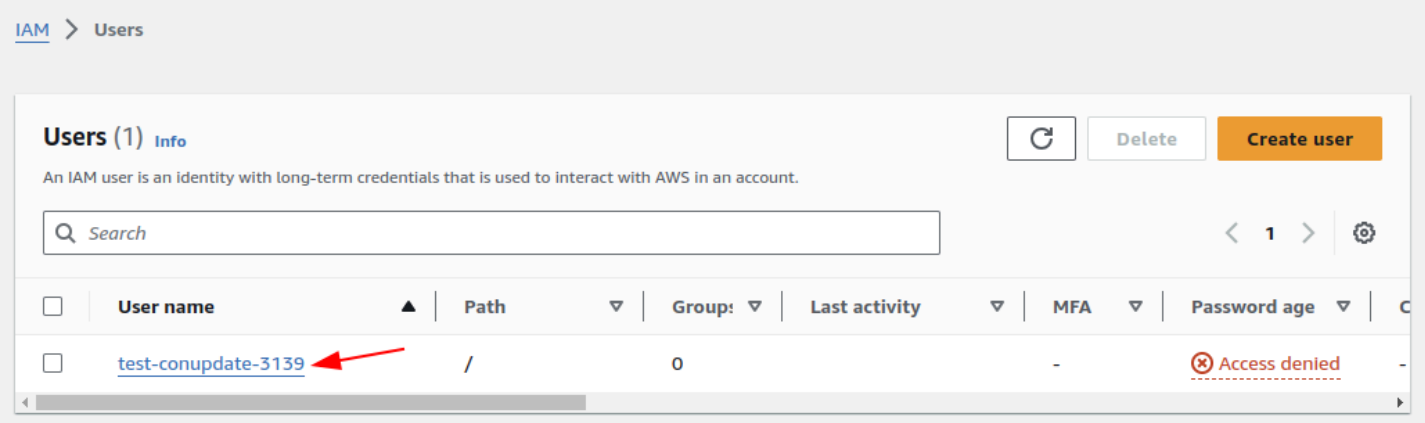
- Set the permissions to the new user by attaching the AWS Managed **AdministratorAccess** policy from the list of existing policies.



Attach the *AdministratorAccess* policy from the list of pre-created policies

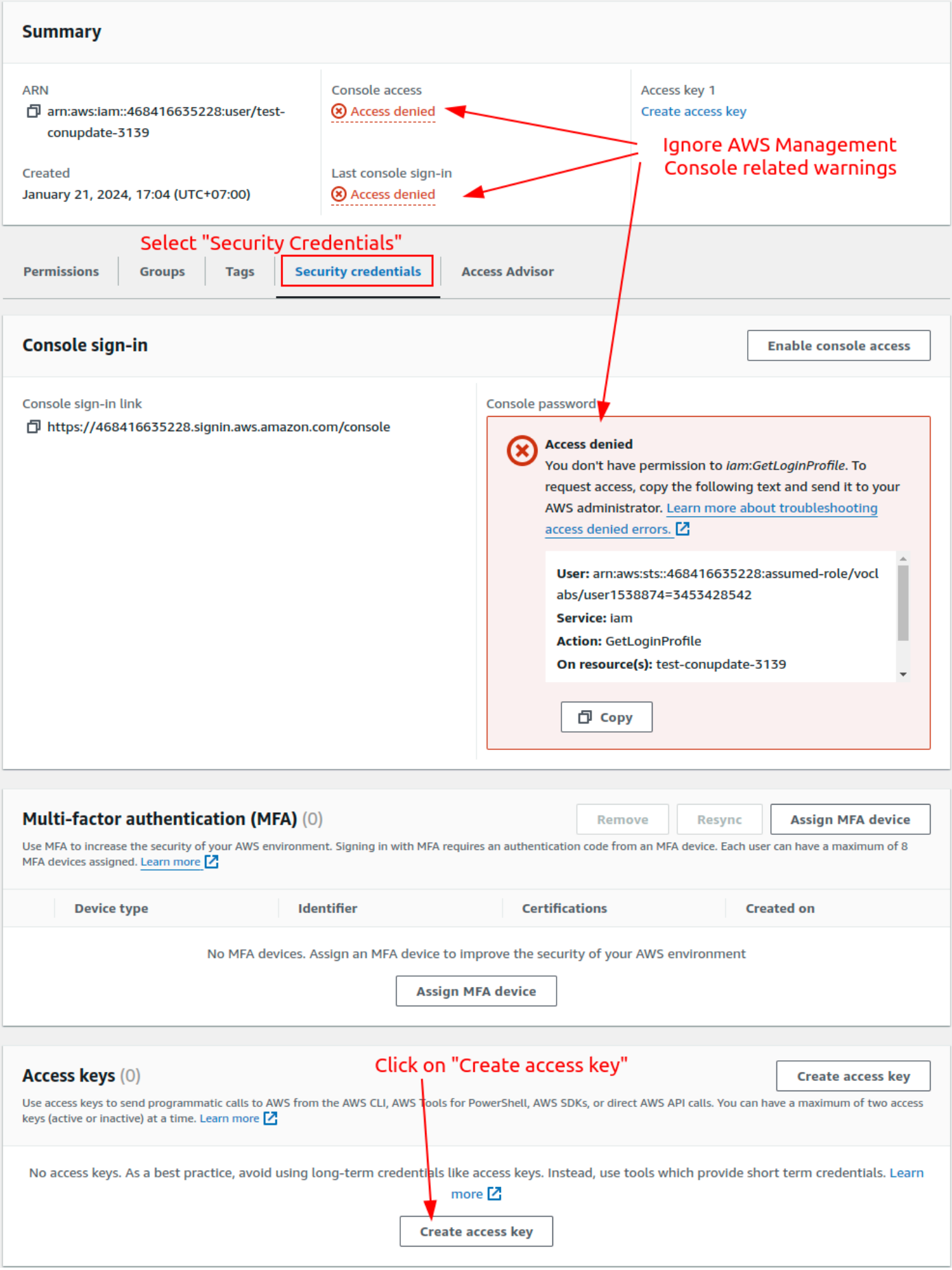
- Provide tags [optional] ([what are tags in AWS?](), review the details of the new user, and finally create the new user.

- After a user is created successfully, click on the User name.



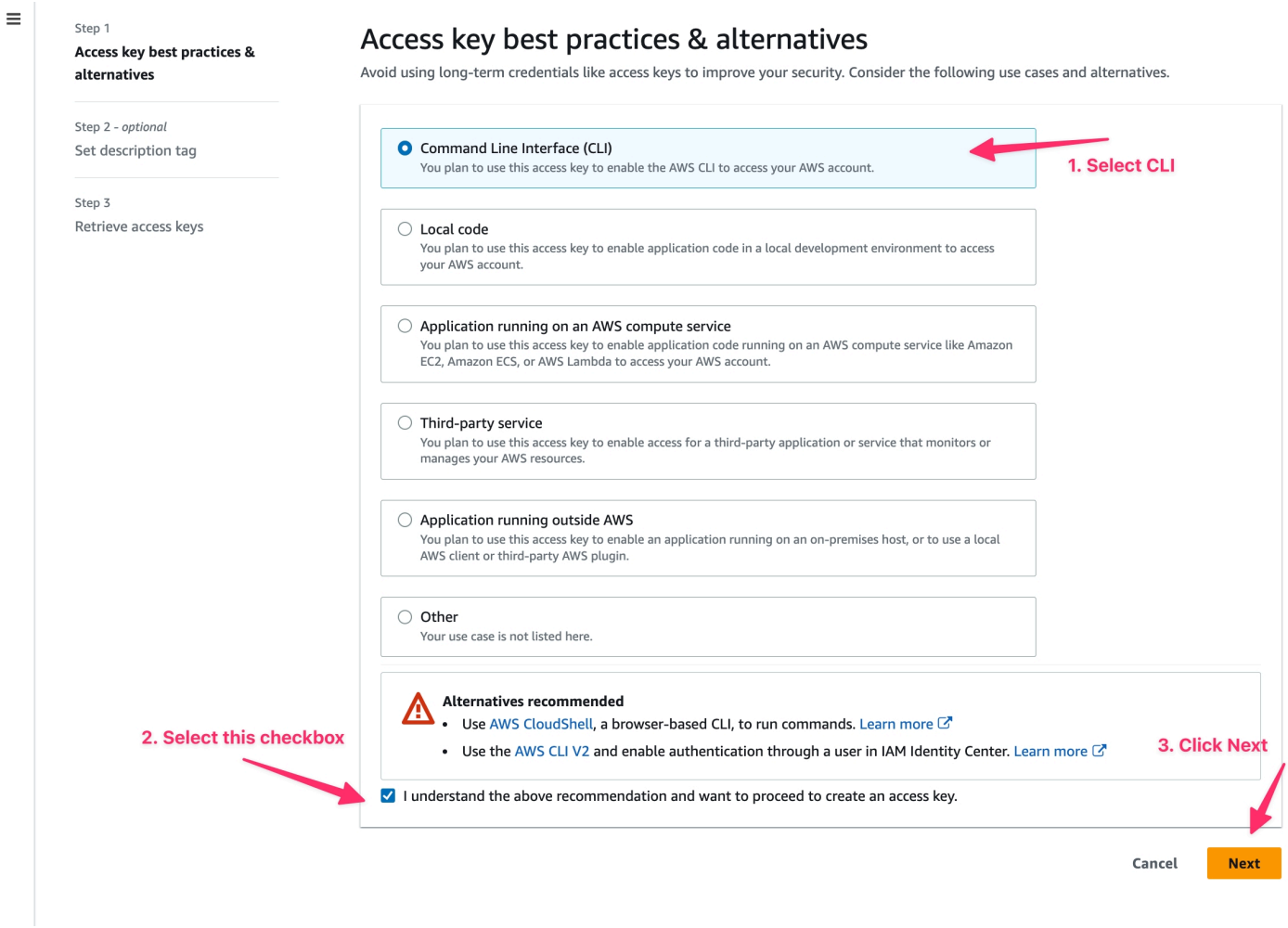
Select the created user.

- Ignore **AWS Management** Console-related warnings. Since you only need programmatic access, this can be ignored. Go to **Security Credentials** and select **Create access key**.



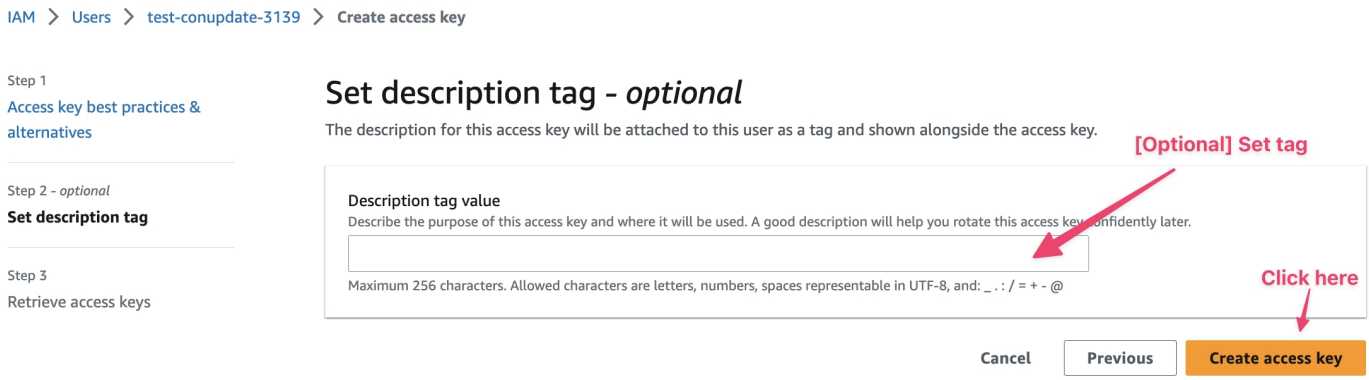
Create Access Key for the user.

- Select **Command Line Interface (CLI)** and click **Next**.



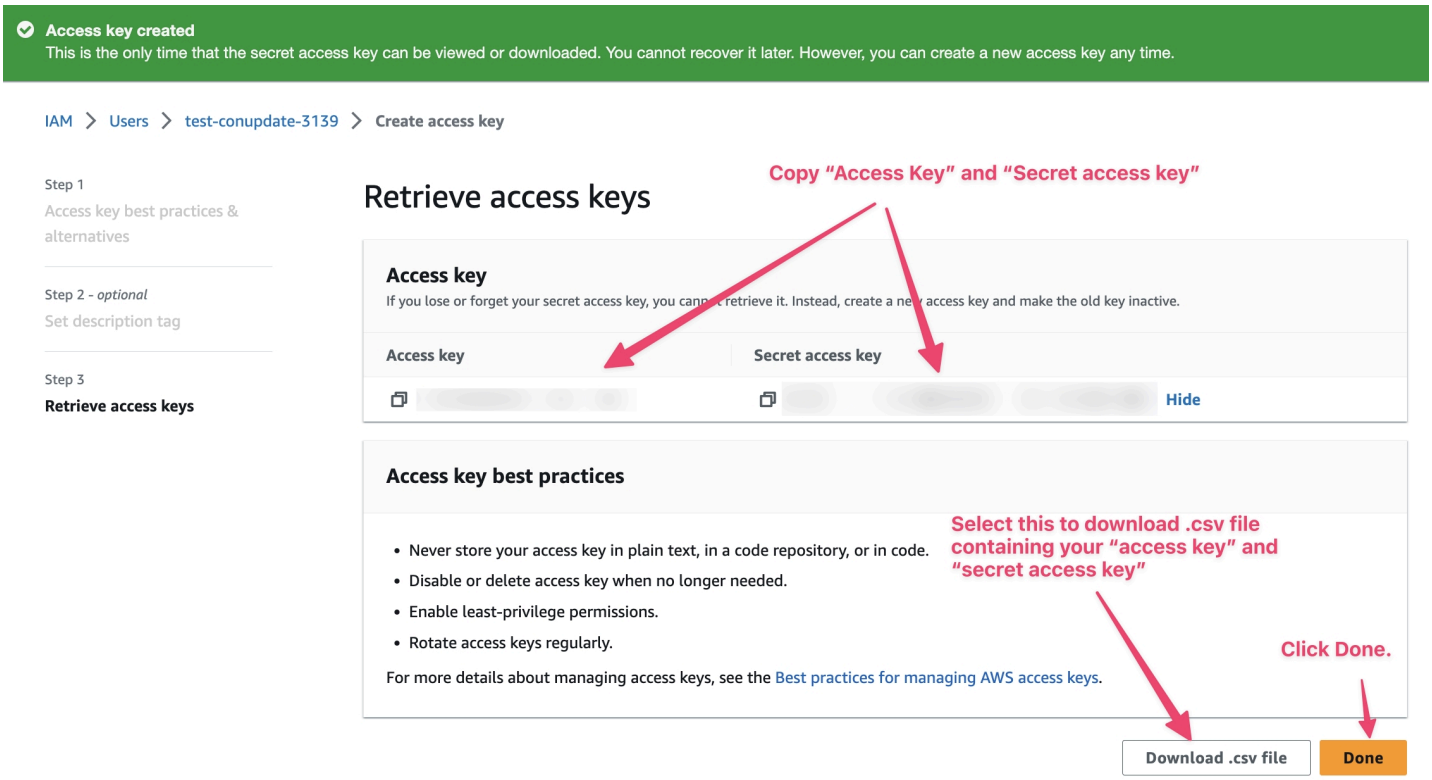
Select *Command Line Interface (CLI)*

- Optional - Set description tag and click **Create access key**.



Optional - Set description tag for the access keys

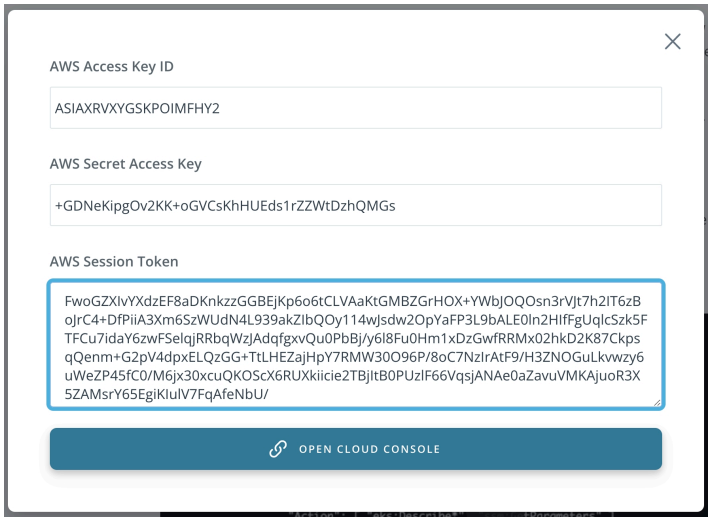
- Copy the created **Access key**, **Secret access key** and store it for later use. You can also download these as a *.csv* file.



Copy *Access Key* and *Secret Access Key*

Note that you can generate a temporary Access key in the classroom as well, as shown in the snapshot below. But, the classroom generated access key is valid for a for a few hours only.

Notice that it has a session token associated with it. The session token needs to be included in your CLI configuration for it to be usable. This will be discussed in more detail in the next section below.



Access key shown in the classroom after clicking on the "OPEN CLOUD GATEWAY" button

Step 3. Configure the AWS CLI

You will need to configure the following four items on your local machine before you can interact with any of the AWS services:

1. **Access key** - It is a combination of an *Access Key ID* and a *Secret Access Key*. Together, they are referred to as *Access key*. You can generate an Access key from the AWS IAM service, and specify the level of permissions (authorization) with the help of *IAM Roles*.
2. **Default AWS Region** - It specifies the AWS Region where you want to send your requests by default.
3. **Default output format** - It specifies how the results are formatted. It can either be a json, yaml, text, or a table.
4. **Profile** - A collection of settings is called a profile. The default profile name is `default`, however, you can create a new profile using the `aws configure --profile new_name` command.

Here are the steps to configure the AWS CLI in your terminal:

- Run the command below to configure the AWS CLI using the *Access Key ID* and a *Secret Access Key* generated in the previous step. If you have closed the web console that showed the access key, you can open the downloaded access key file (.csv) to copy the keys later.

```
aws configure
```

If you already have a profile set locally, you can use `--profile <profile-name>` option with any of the AWS commands above. This will resolve the conflict with the existing profiles set up locally. Next, use the following values in the prompt that would appear:

Important

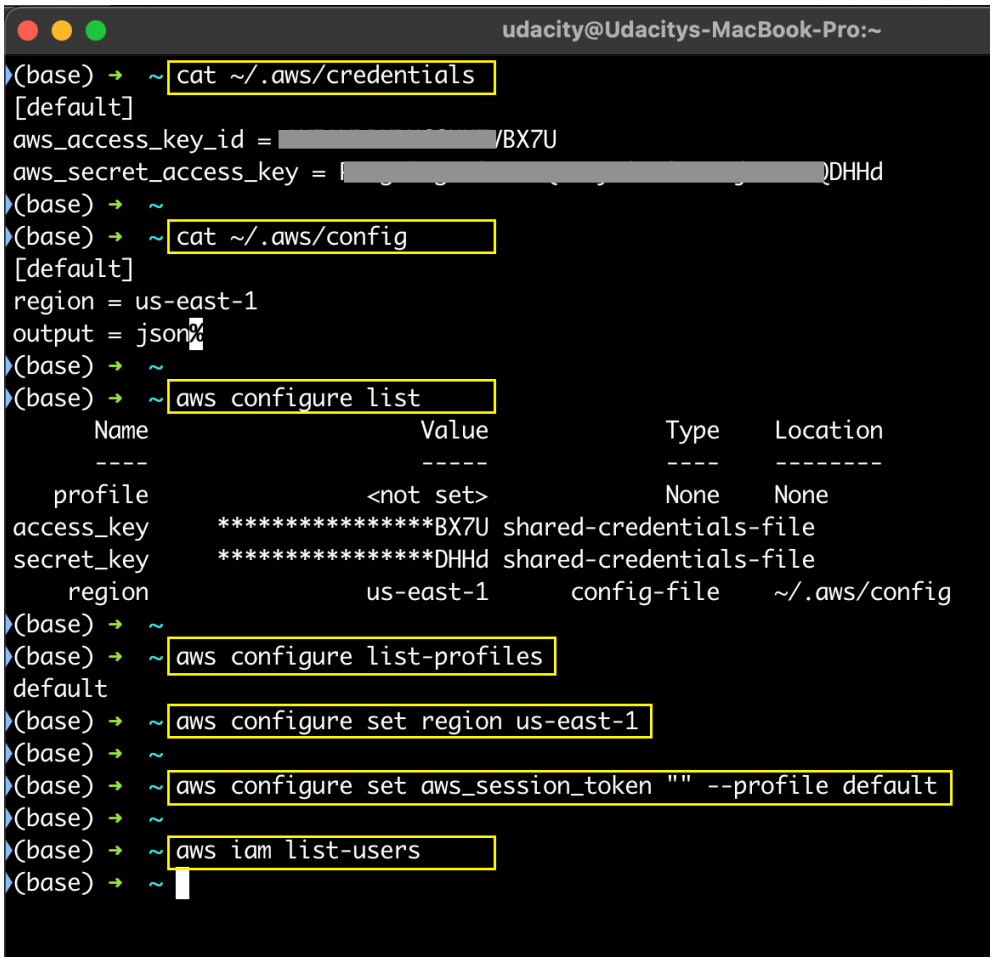
If you are using the Udacity-generated Access key, you should set the `aws_session_token` parameter.

```
aws configure set aws_session_token "XXXXXXXX"
```

where `"XXXXXXXX"` is the **AWS Session Token** copied from the classroom after clicking on the "OPEN CLOUD GATEWAY" button.

If you are using the Access key of an Admin IAM user, you do not need to set this parameter.

- The commands above will store the access key in a default file `~/.aws/credentials` and store the profile in the `~/.aws/config` file. Upon prompt, paste the copied access key (access key id and secret access key). Enter the default region as `us-east-1` and output format as `json`. You can verify the saved config using:



Mac/Linux: A successful configuration

The above steps should be sufficient for setting up your AWS CLI locally, but if you still can't run AWS commands, you may try following the steps below.

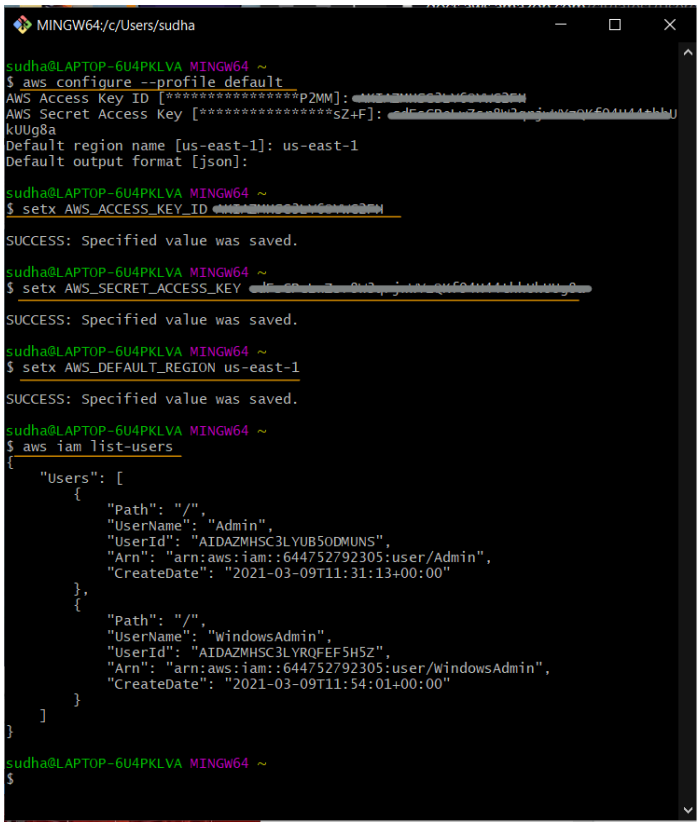
- **Mac/Linux users** Let the system know that your sensitive information is residing in the .aws folder

```
export AWS_CONFIG_FILE=~/.aws/config
export AWS_SHARED_CREDENTIALS_FILE=~/.aws/credentials
```

- **Windows users with GitBash only** You will have to set the environment variables. Run the following commands in your GitBash terminal:

```
setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
setx AWS_DEFAULT_REGION us-east-1
```

Replace the access key ID and secret, as applicable to you. Windows users using WSL do not need this step, they will follow all steps as if they are Linux users.



Windows: Successful configuration using the GitBash terminal

Step 4. Run your first AWS CLI command

- Check the successful configuration of the AWS CLI, by running either of the following AWS command:

```
# If you've just one profile set locally
aws iam list-users
# If you've multiple profiles set locally
aws iam list-users --profile <profile-name>
```

The output will display the details of the recently created user:

```
{
"Users": [
    {
        "Path": "/",
        "UserName": "Admin",
        "UserId": "AIDAZMXYZ3LY2BNC5ZM5E",
        "Arn": "arn:aws:iam::388752792305:user/Admin",
        "CreateDate": "2021-01-28T13:44:15+00:00"
    }
]
}
```

Troubleshoot

If you are facing issues while following the commands above, refer to the detailed instructions here -

1. **Configuration basics**
2. **Configuration and credential file settings**

3. **Environment variables to configure the AWS CLI**

Updating the specific variable in the configuration

In the future, you can set a single value, by using the command, such as:

```
# Syntax
# aws configure set <varname> <value> [--profile profile-name]
 aws configure set default.region us-east-1
```

It will update only the region variable in the existing default profile.